Docket No : POU920010001US1

Inventor : **Mark A. Nelson et al**
Title : **Method, System and Storage Medium for Determining Trivial Keyboard Sequences of Proposed Passwords**

APPLICATION FOR UNITED STATES

LETTERS PATENT

"Express Mail" Mailing Label No.: ET089520818US
Date of Deposit: January 23, 2002

I hereby certify that this paper is being deposited with the United States Postal Service as "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231

Name: **Mariann Kelly**

Signature: Mariann Kelly

INTERNATIONAL BUSINESS MACHINES CORPORATION

METHOD, SYSTEM, AND STORAGE MEDIUM FOR DETERMINING TRIVIAL KEYBOARD SEQUENCES OF PROPOSED PASSWORDS

BACKGROUND

[0001] This invention relates generally to password security systems, and more particularly, the present invention relates to a method, system and storage medium for determining trivial keyboard sequences of proposed passwords.

[0002] Secure computer network systems rely on security mechanisms to protect the integrity of the applications and information stored therein. Password-based mechanisms are the most common of these security systems and involve the selection of a string of alphanumeric characters that can be assigned either by a system administrator or self-assigned by a system user. The effectiveness of these security mechanisms depend, in part, upon the ability of system users to maintain discreet password usage over time and throughout the duration of network access. One difficulty, however, lies in the struggle to create a balance between the need for providing easily-remembered passwords against the security risks in doing so. Common words and phrases are vulnerable to external and internal attack. Various software programs exist that attempt to gain access to computer systems via systematic login attempts using common words and phrases (also referred to as weak passwords) until a match is found. Selecting non-obvious passwords may not necessarily solve the security problem because they are subject to compromise when password owners who have trouble remembering them resort to keeping written notes with the password. The chances of the written password getting into the wrong hands becomes a risk to the security of the network system.

[0003] Virtually every operating system environment provides some controls which attempt to ensure the quality of passwords. Types of controls include: requiring periodic changes of passwords, preventing password re-use, defining minimum length standards for passwords, adopting semantic content restrictions (e.g., passwords may not contain any three-character abbreviation for the months of the year, or a new password may not contain any three sequential

characters that are the same as in the existing password), as well as trivial keyboard sequences (e.g., "qwerty").

[0004] Various solutions have been devised to reduce or eliminate the problem of weak passwords (e.g., those utilizing common words or trivial keyboard sequences). Known solutions directed to weak passwords relate to password evaluation systems that evaluate the proposed password or substrings of the password against a 'dictionary' or database of known 'bad' password sets, either via a statistical method or a hashing table. These solutions are somewhat limited in that their success depends heavily on the quality and comprehensiveness of the 'bad' password sets. They are also time consuming since proposed passwords and/or its substrings must be each compared against voluminous database entries. Also, there is no guarantee a match will be found for certain common words. Trivial keyboard passwords may be particularly immune from implementation of these solutions because they do not conform to general 'dictionary'-based requirements but instead use computer keyboard sequences. Determining keyboard triviality in prior art systems generally involves checking the password against known character strings, that are stored in a data file. This is a time-consuming process as large database searches are required and all variations of keyboard sequences would be necessary to ensure success.

[0005] What is needed is quicker and more direct way to determine trivial keyboard sequences of proposed passwords.

BRIEF SUMMARY

[0006] An exemplary embodiment of the invention relates to a method, system, and storage medium for determining trivial keyboard sequences of a proposed password. The system comprises a user system and associated keyboard input device; a server in communication with the user system via a communications link; a data storage device coupled to the server, housing a database including a keyboard profile for specifying a physical layout of character and function keys on the computer keyboard input device; a master password database including a user account; and a password verification mechanism executable by the server. Upon execution, the

password verification mechanism performs an algorithm on the proposed password and determines triviality according to criteria specified in the algorithm. The physical layout of character and function keys is specified by a graphical representation of the computer keyboard input device; an X axis horizontally spanning the graphical representation; and a Y axis vertically spanning the graphical representation. Each of the character and function keys is assigned a unique data coordinate set identifying positional placement values. The network system also includes an identifier assigned to the keyboard profile indicating manufacturer and model data. A keyboard profile, a method, and storage medium for determining triviality of a proposed password are also included.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Referring now to the drawings wherein like elements are numbered alike in the several FIGURES:

[0008] FIG. 1 depicts a computer network system upon which the password verification mechanism is implemented in an exemplary embodiment;

[0009] FIG. 2 is a sample keyboard profile with assigned 'X' and 'Y' axes and corresponding data points for indicating keys located on a keyboard; and

[0010] FIG. 3 is a flowchart describing the process of determining trivial keyboard sequences utilizing the password verification mechanism in an exemplary embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0011] The password verification invention addresses the issue of determining trivial keyboard sequences used for proposed password requests. A computer keyboard is represented as a two-dimensional graph, where the X-axis represents the placement of keys in a column of a keyboard, and the Y-axis represents the placement of keys in a row of the keyboard. The

password verification mechanism performs a mathematical algorithm on the proposed password according to its assigned data points in order to determine triviality. A standard parameter is set which is used to compare the values derived from the execution of the mathematical algorithm in order to assess acceptable distances between proposed password characters as displayed on the keyboard. If the values are acceptable, the process is finished and the password is approved.

[0012] In an exemplary embodiment, the password verification mechanism is implemented on a computer network system such as that depicted in FIG. 1. Network system 100 includes a user system 102 coupled to a server 104 via a communications link 106. System 100 may be a central facility for a business enterprise which executes the password verification mechanism (e.g., regional/global hub facility) or may itself comprise the entire business enterprise. Additional facilities or hubs may be included in system 100 in order to realize the advantages of the invention. Such might be the case where the business enterprise implementing the password verification mechanism is a large global enterprise with offices, sites, and/or distribution centers dispersed around the world. User system 102 and administrator system 114 may be general-purpose computers such as a personal computer (PC), laptop, or handheld appliance that include a processor, memory, computer keyboard input devices, and suitable output devices. User system 102 and administrator system 114 execute one or more computer programs for carrying out the processes described herein. It should be noted that any number of user systems and administrator systems may be utilized by network system 100. Alternatively, user system 102 and/or administrator system 114 may employ applications stored on server 104 wherein user system 102 and administrator system 114 operate as 'dumb' clients and server 104 carries out the processes described herein with respect to the password verification mechanism. Typical users of user system 102 may include management, support staff, and other representatives of the business enterprise. Typical users of administrator system 114 may include security personnel, information technology (IT) specialists, systems maintenance personnel, etc. Communications link 106 may comprise a local area network (LAN), a wide area network (WAN), or other network configuration known in the art. Further, link 106 may include wireless connections, radio-based communications, telephony-based communications, and other network-based communications. For purposes of illustration, however, communications link 106 is a LAN.

[0013] Server 104 may be executing suitable web server software designed to accommodate various forms of network communications, including voice, video, and text. Server 104 may also be running e-mail and groupware applications typically found in a business environment. Server 104 executes database management software and security software for assisting users of the password verification mechanism in establishing and maintaining password accounts. Security features may be achieved via a firewall or similar security device for limiting access to network system 100 to those users possessing proper access permissions. For instance, an administrator at system 114 may have access to the entire system and have authority to modify portions of the system. By contrast, a low level employee on user system 102 may have the ability to execute programs but not alter the applications or data stored in data storage device 108. It is understood that more than one server may be used

[0014] Server 104 may be coupled to a data storage device 108 via communications link 106. Data storage device 108 is any form of mass storage device configured to read and write database type data maintained in a file store (e.g., a magnetic disk data storage device). Data storage device 108 may be logically addressable as a consolidated data source across a distributed environment such as a network system. The implementation of local and wide-area database management systems to achieve the functionality of data storage device 108 will be readily understood by those skilled in the art. Information stored in data storage device 108 may be retrieved and manipulated by database management software executed by server 104.
Data storage device 108 contains a variety of information and databases related to the password verification mechanism as well as proprietary information desired by network system 100. Keyboard profiles database 110 houses keyboard profiles related to user systems utilized by the business enterprise. Keyboard profiles define the layout of character and function keys of a computer keyboard for purposes of assigning data coordinates. Keyboard profiles for a variety of computer models are stored in database 110. An example of a keyboard profile is illustrated in FIG. 2.

[0015] Master password database 112 stores current validation information for user accounts and may also store keyboard-identifying information related to the system devices assigned to password users. For example, user system's 102 password account may be tagged with keyboard identifying information relating to the keyboard profile that coincides with the user's computer. Other databases may be included in network system 100 as desired by the business enterprise. Data stored in data storage device 108 is accessed by server 104 during presentation of the password verification program to user system 102 and/or administrator system 114. It will be understood that data storage device 108 and server 104 may comprise one server/storage unit and that multiple server/storage units may be employed by network system 100 in order to realize the advantages of the invention.

[0016] Fig. 2 illustrates a sample keyboard profile for an IBM ThinkPad 570(TM). A graphical representation of a computer keyboard is displayed indicating actual physical location of character and functions keys as they appear on an actual keyboard. An X-axis spans the keyboard profile horizontally and includes assigned data points X1 - X15. A Y-axis spans the keyboard profile vertically and includes assigned data points Y0 - Y6. Thus, the coordinates of a password associated with a keyboard profile includes the following data.

PASSWORD	DATA COORDINATES
First letter	(X1, Y1)
Second letter	(X2, Y2)
Third letter	(X3, Y3)
...	
nth letter	(Xn, Yn)

[0017] The data coordinates for letter 'J' of the keyboard profile of FIG. 2 would be (9, 2). The utility of these assigned data points will be described further herein.

[0018] Fig. 3 illustrates a flowchart describing the process of determining trivial keyboard sequences of proposed passwords using the password verification mechanism. A user at user system 102 accesses the password verification mechanism at step 302. The user enters a proposed password request at step 304. The password verification mechanism accesses master password database 112 and checks the proposed password against existing password quality rules, such as minimum length, semantic content, and reuse in database 112 at step 306. If the mechanism finds an inappropriate password (step 308) it redirects the user to select a different password (step 304). If the password has passed the first acceptability test, flow proceeds to step 310 whereby the password verification mechanism is invoked. The password verification mechanism accesses keyboard profile database 110 and retrieves the keyboard profile associated with the user system requesting the password at step 312. This may be accomplished using various techniques. The mechanism may receive an automatic signal from the requesting user system indicating the name, brand, model, etc. of the keyboard/user system in use. Of course, this step may not be required where only one keyboard type is utilized by computer network system 100. The mechanism may also be configured to provide the user with a listing of keyboard/system types available whereby the user selects the appropriate item on the list.

[0019] The password verification mechanism executes an algorithm on the proposed password utilizing one or more of three formulas designed to minimize the occurrence and assignment of trivial keyboard passwords. The first two formulas verify that the key strokes associated with the proposed password are not on the same row and column, and the third formula assures a diverse key stroke pattern. If the first formula results in a failure, it is not necessary to proceed with the execution of the second formula and the process ends. Likewise, if the second formula results in a failure, it is not necessary to proceed with the execution of the third formula.

[0020] It will be noted that proposed passwords that contain mixed case values may be folded to a single case before the validation mechanism is invoked.

[0021] For purposes of illustration, a first proposed password provided by user system 102 is 'qwerty'. Utilizing the keyboard profile of FIG. 2, this password selection would result in data points (3,3), (4,3), (5,3), (6,3), (7,3), and (8,3).

[0022] A first formula (F1) is executed at step 314, checking for vertical keyboard sequences (also referred to as 'vertical triviality').

F1: $(\triangle X1 + \triangle X2 + ... + \triangle xn\text{-}1)/(n-1) > 0$

[0023] The following conditions apply to all formulas where 'n' is the length of the password.

$\triangle X1$ equals the difference between X1 and X2.

$\triangle X2$ equals the difference between X2 and X3.

...

$\triangle Xn$ equals the difference between Xn+1 and Xn.

[0024] In general, let $1 <= m < n$ whereby $\triangle Xm$ is the absolute value of the difference between the X coordinate of letter m and letter m + 1 (e.g., $\triangle Xm = \triangle Xm - \triangle Xm+1$) and $\triangle Ym$ is the absolute value of the difference between the Y coordinate of letter m and letter m+1 (e.g., $\triangle Ym = \triangle Ym - \triangle Ym+1$)

[0025] S is a system installation parameter and represents the mean distance between character keys used for comparisons. For purposes of illustration, S has been set at '2'.

[0026] For the 'qwerty' password example, the first formula applied to it data coordinates results as follows.

F1: $(1 + 1 + 1 + 1 + 1)/ 5 = 1$

[0027] The indicated result of '1' is a valid sequence (step 316) and so the process continues at step 318 where a second formula of the algorithm is executed. Formula 2 verifies horizontal keyboard sequences (also referred to as 'horizontal triviality').

F2: $(\triangle Y1 + \triangle Y2 + ... + \triangle Yn\text{-}1)/(n\text{-}1) > 0$

[0028] With values of proposed password qwerty plugged in to F2, the following results are indicated.

F2: $(0 + 0 + 0 + 0 + 0)/5 = 0$ The indicated result of '0' is an invalid sequence (step 320) and causes a failure and so the process returns to step 304 whereby the mechanism directs the user to provide an alternative password.

[0029] Because the second formula failed, the mechanism will not need to initiate formula three. For purposes of illustration, a second password 'Ap_5ple' is provided that will facilitate the description of the execution of the third formula.

[0030] Assuming for purposes of illustration that execution of F1 and F2 resulted in a valid sequence, a third formula (F3) is initiated at step 322 as follows.

F3: $(\triangle X1 + \triangle Y1 + \triangle X2 + \triangle Y2 + ... + \triangle xn\text{-}1 + \triangle yn\text{-}1)/(2(n\text{-}1)) >= S$

or $(F1 + F2)/2 >= S$ F3: $(1 + 0)/2 = .5$ or

F3: $(1 + 9 + 1 + 0 + 0 + 6 + 1 + 6 + 1 + 1 + 1 + 6)/2 * 6 = 33/12 = 2.75$

[0031] Since the system installation parameter is set at 2, this sequence would pass. The installation parameter number reflects the average distance between key strokes.

[0032] Once all three formulas have been validated (step 324), the mechanism transmits an acceptance of the proposed password to the user system 102 and/or administrator system 114 at step 326. The mechanism then updates password database 112 to reflect the new password at step 328.

[0033] As described above, the present invention can be embodied in the form of computer-implemented processes and apparatuses for practicing those processes. The present invention can also be embodied in the form of computer program code containing instructions embodied in tangible media, such as floppy diskettes, CD-ROMs, hard drives, or any other computer-readable storage medium, wherein, when the computer program code is loaded into and executed by a computer, the computer becomes an apparatus for practicing the invention. The present invention can also be embodied in the form of computer program code, for example, whether stored in a storage medium, loaded into and/or executed by a computer, or transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via electromagnetic radiation, wherein, when the computer program code is loaded into and executed by a computer, the computer becomes an apparatus for practicing the invention. When implemented on a general-purpose microprocessor, the computer program code segments configure the microprocessor to create specific logic circuits.

[0034] While preferred embodiments have been shown and described, various modifications and substitutions may be made thereto without departing from the spirit and scope of the invention. Accordingly, it is to be understood that the present invention has been described by way of illustration and not limitation.